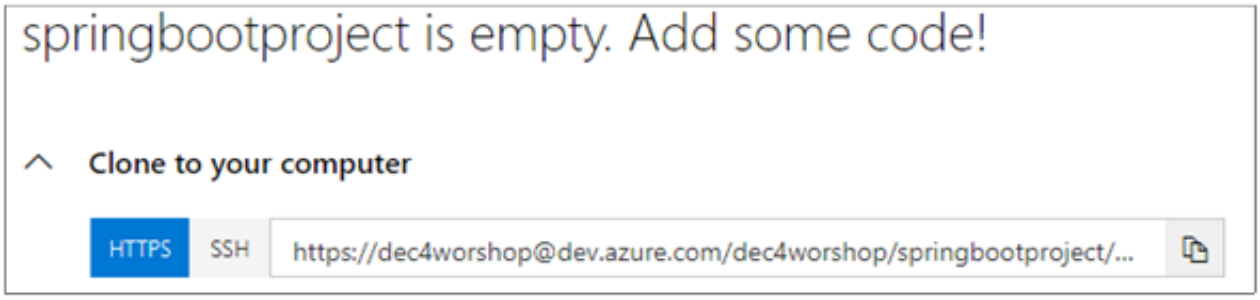
Before you get started, make sure you have a project created in dev.azure.com. Here, the project created is “springbootproject”.

Step 1:

Navigate to the path where you want to clone the repo.

Copy the clone URL from the “springbootproject”.



Step 2:

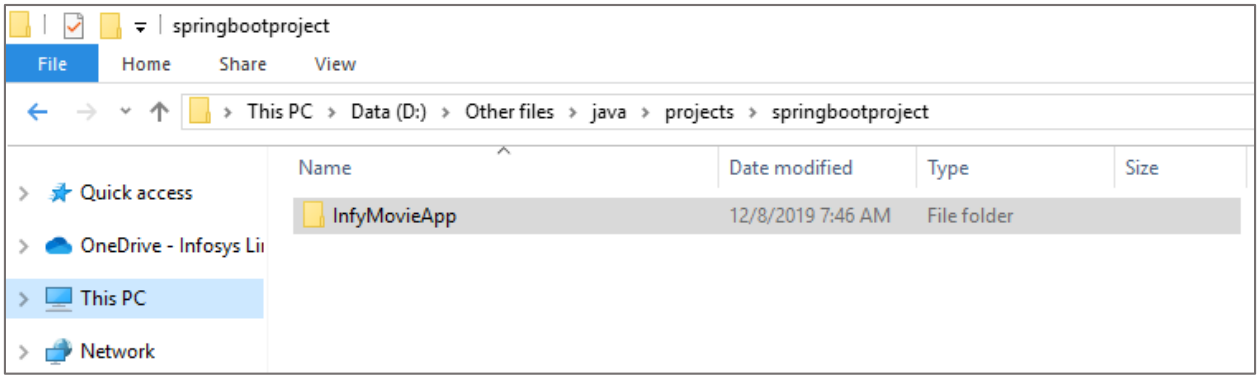
Clone the repo by providing the below git clone command.

```
D:\Other files\java\projects>git clone https://dec4worshop@dev.azure.com/dec4worshop/springbootproject/_git/springbootpr
oject
Cloning into 'springbootproject'...
warning: You appear to have cloned an empty repository.
```

Step 3:

InfyMovieApp.zip

Now navigate to cloned repository, Copy and paste the java spring boot service provided to you into the cloned repository.



Step 4:

Provide the below command to add the project to the repository.

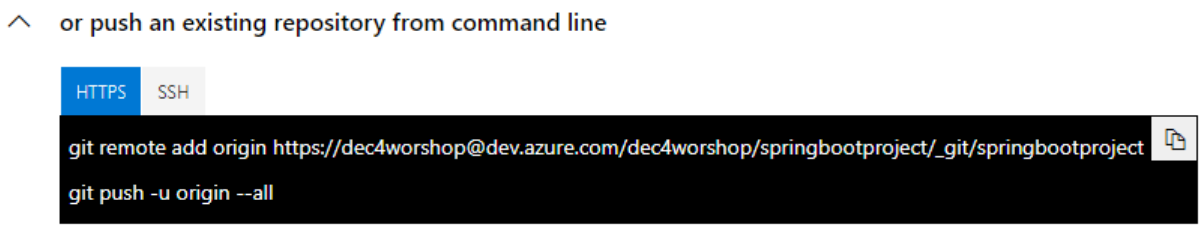
```
D:\Other files\java\projects>git add .
```

Now commit.

```
D:\Other files\java\projects>git commit -m "First commit"
[master (root-commit) 02c383d] First commit
 Committer: Juliet R. Rajan <Juliet_Rajan@ad.infosys.com>
```

Step 5:

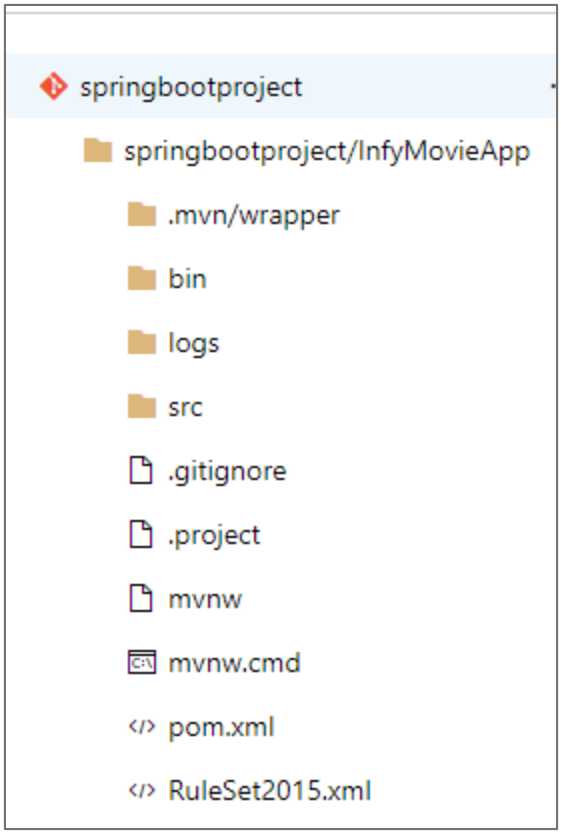
Copy the below commands from your remote repo and execute it to push the local changes to the remote repo.


or push an existing repository from command line
HTTPS
SSH
git remote add origin https://dec4worshop@dev.azure.com/dec4worshop/springbootproject/_git/springbootproject
git push -u origin --all

```
D:\Other files\java\projects>git remote add origin https://dec4worshop@dev.azure.com/dec4worshop/springbootproject/_git/
springbootproject

D:\Other files\java\projects>git push -u origin --all
fatal: NullReferenceException encountered.
   Object reference not set to an instance of an object.
Password for 'https://dec4worshop@dev.azure.com':
fatal: NullReferenceException encountered.
   Object reference not set to an instance of an object.
Enumerating objects: 85, done.
Counting objects: 100% (85/85), done.
Delta compression using up to 4 threads
Compressing objects: 100% (56/56), done.
Writing objects: 100% (85/85), 95.48 KiB | 2.08 MiB/s, done.
Total 85 (delta 15), reused 0 (delta 0)
remote: Analyzing objects... (85/85) (84 ms)
remote: Storing packfile... done (157 ms)
remote: Storing index... done (33 ms)
To https://dev.azure.com/dec4worshop/springbootproject/_git/springbootproject
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
```

You will see the pushed changes in the remote repo as below.

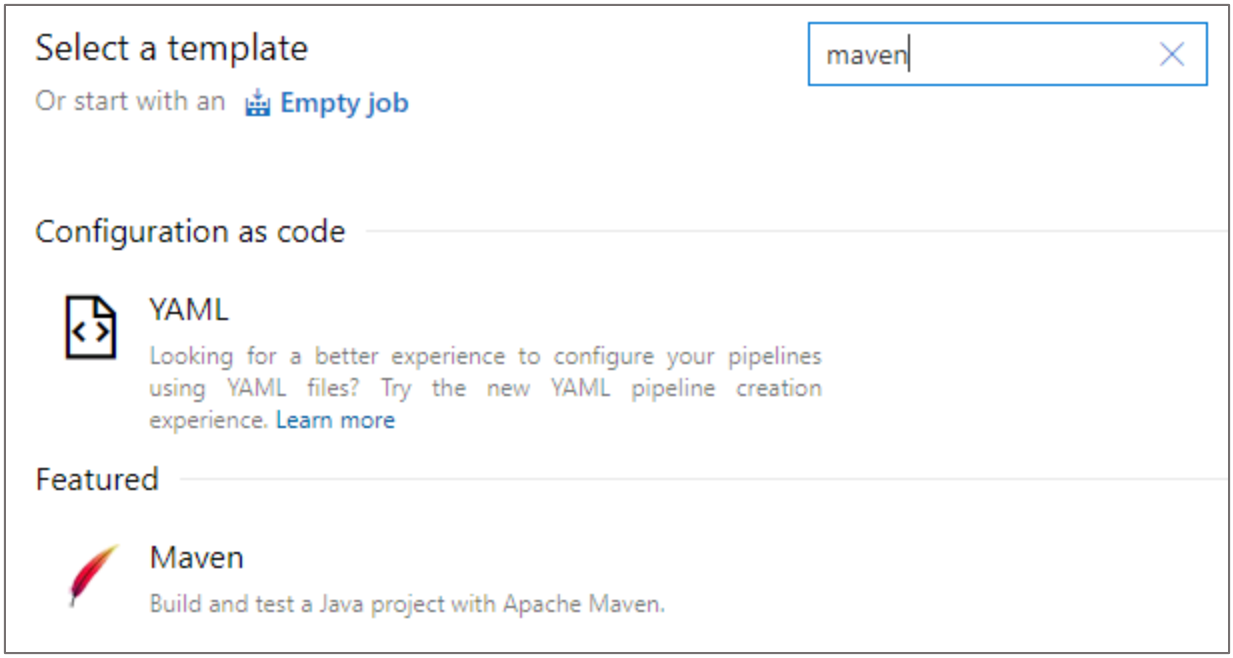


Step 6:

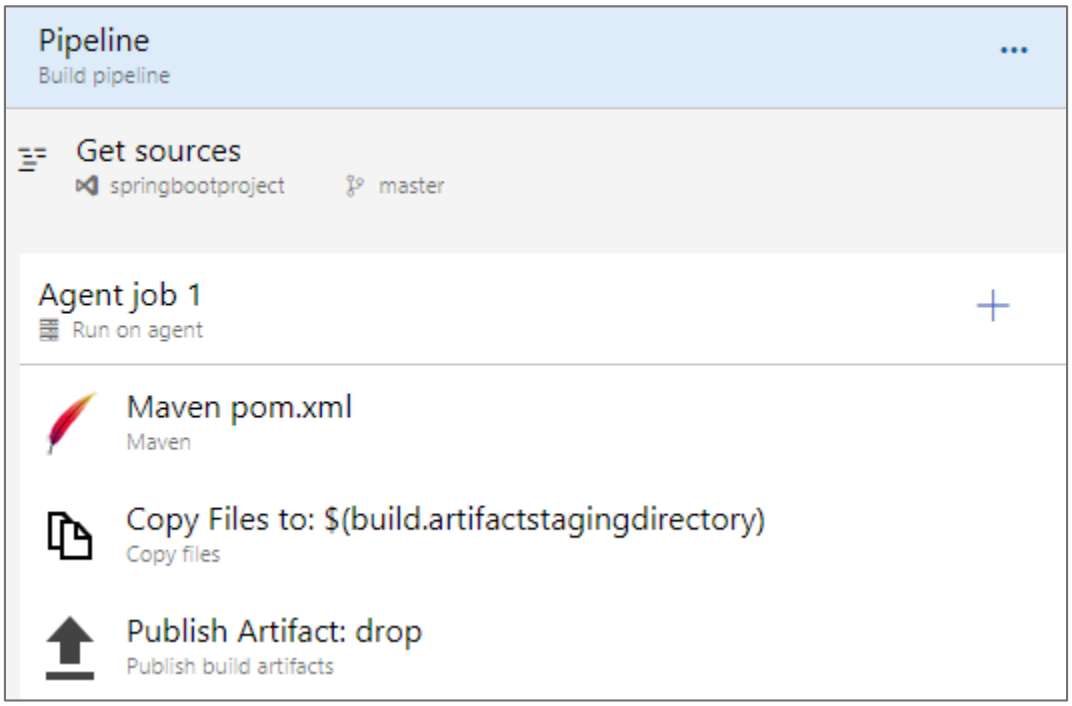
Navigate to pipeline section.

To configure, create a new build pipeline.

Choose a new maven build template.

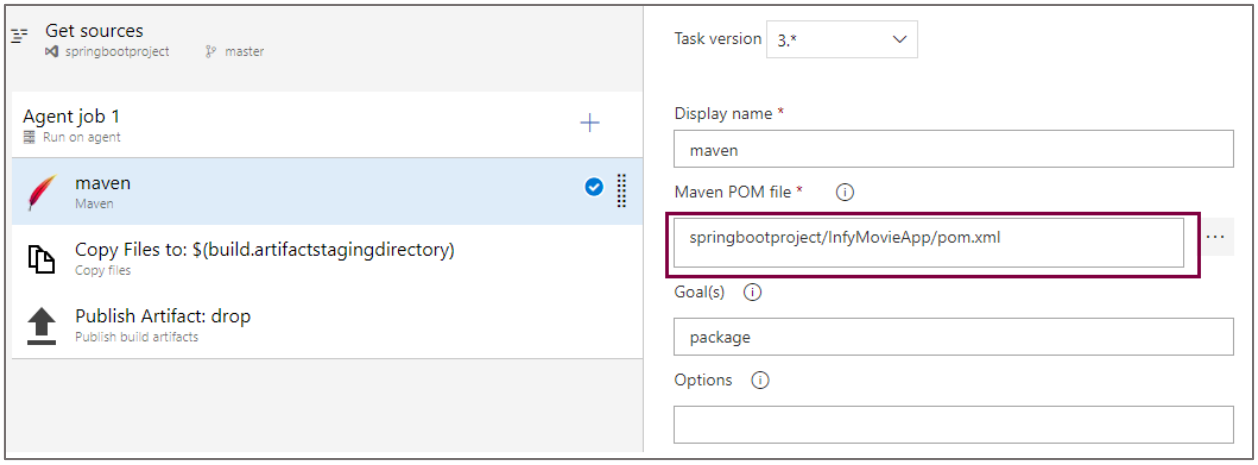

Select a template
maven
Or start with an Empty job
Configuration as code
YAML
Looking for a better experience to configure your pipelines using YAML files? Try the new YAML pipeline creation experience. Learn more
Featured
Maven
Build and test a Java project with Apache Maven.

You will see the build pipeline consists of 3 build agents.

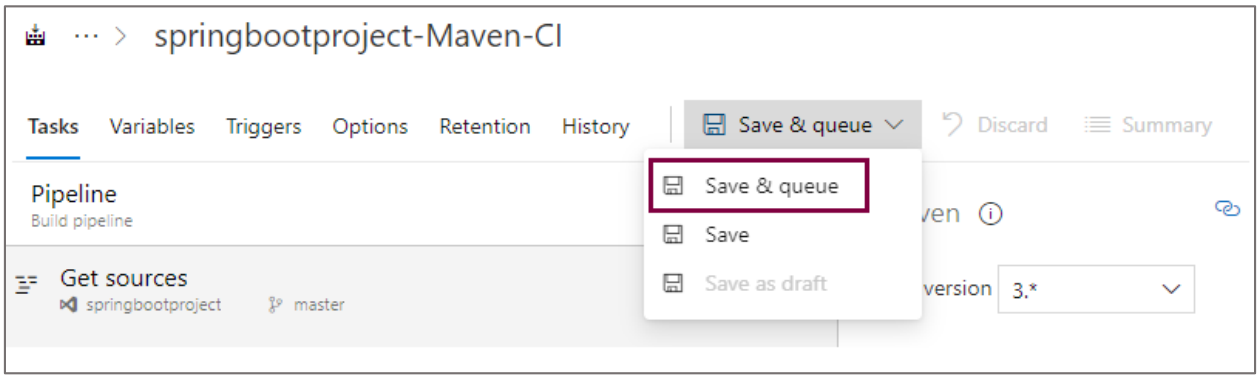


Step 7:

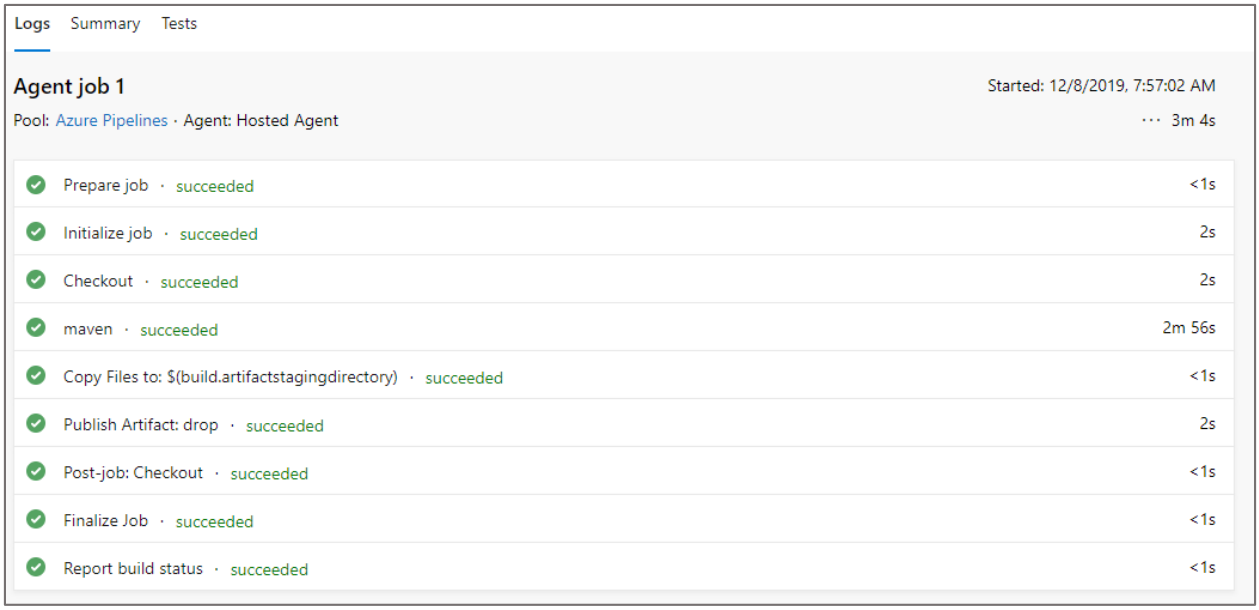
Click on Maven pom.xml. Choose the pom.xml.



Click on Save and Queue.



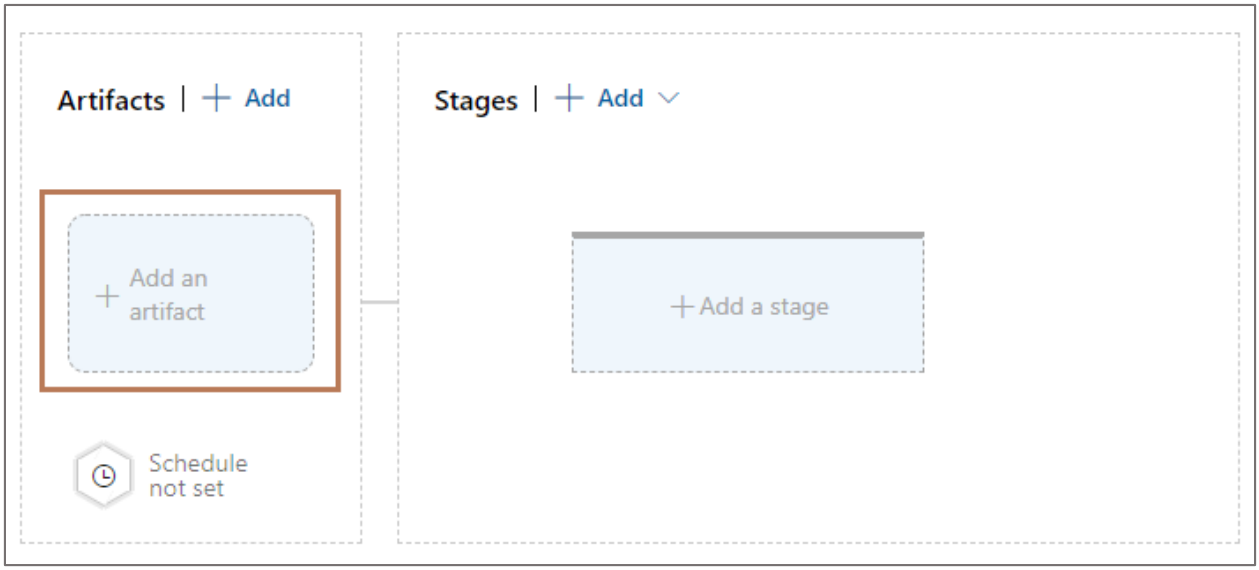
The build is completed successfully.


Logs
Summary
Tests
Agent job 1
Started: 12/8/2019, 7:57:02 AM
Pool: Azure Pipelines · Agent: Hosted Agent
3m 4s
Prepare job · succeeded <1s
Initialize job · succeeded 2s
Checkout · succeeded 2s
maven · succeeded 2m 56s
Copy Files to: $(build.artifactstagingdirectory) · succeeded <1s
Publish Artifact: drop · succeeded 2s
Post-job: Checkout · succeeded <1s
Finalize Job · succeeded <1s
Report build status · succeeded <1s

Step 8:

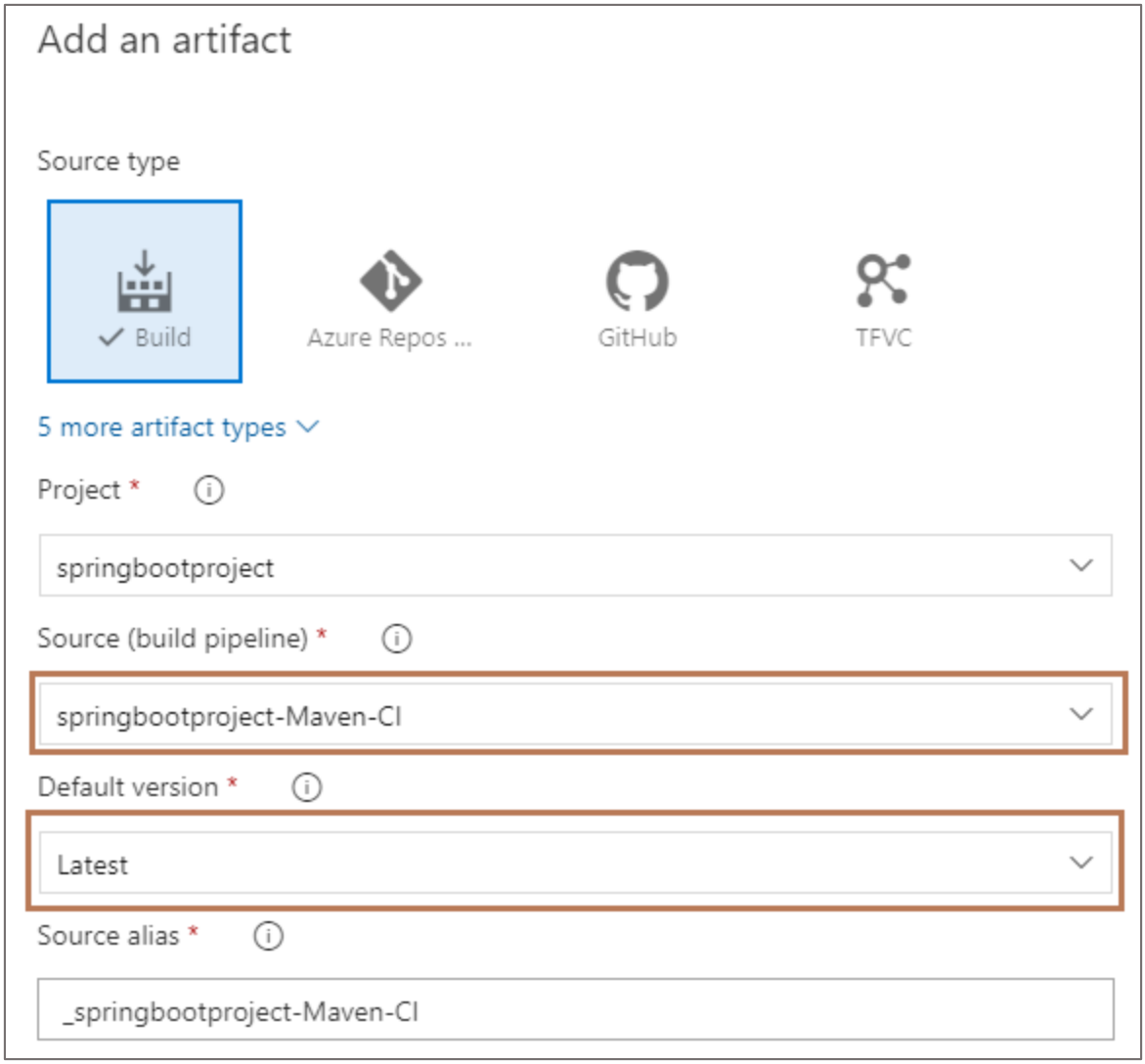
Configure the release pipeline.

Click on "Add an artifact".



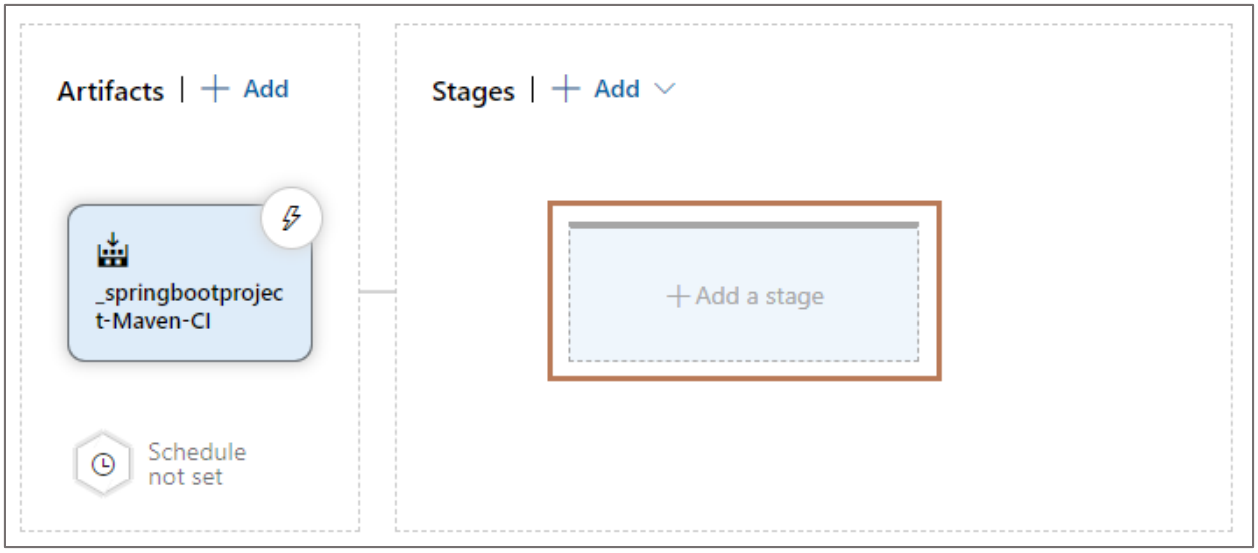
Step 9:

Choose the build pipeline you created earlier and the version of the build.



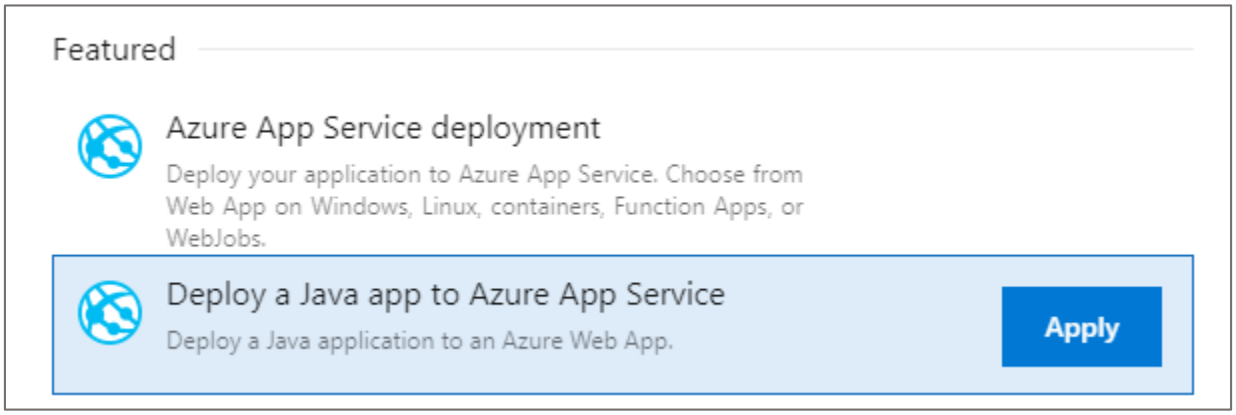
Step 10:

Click on 'Add a stage".



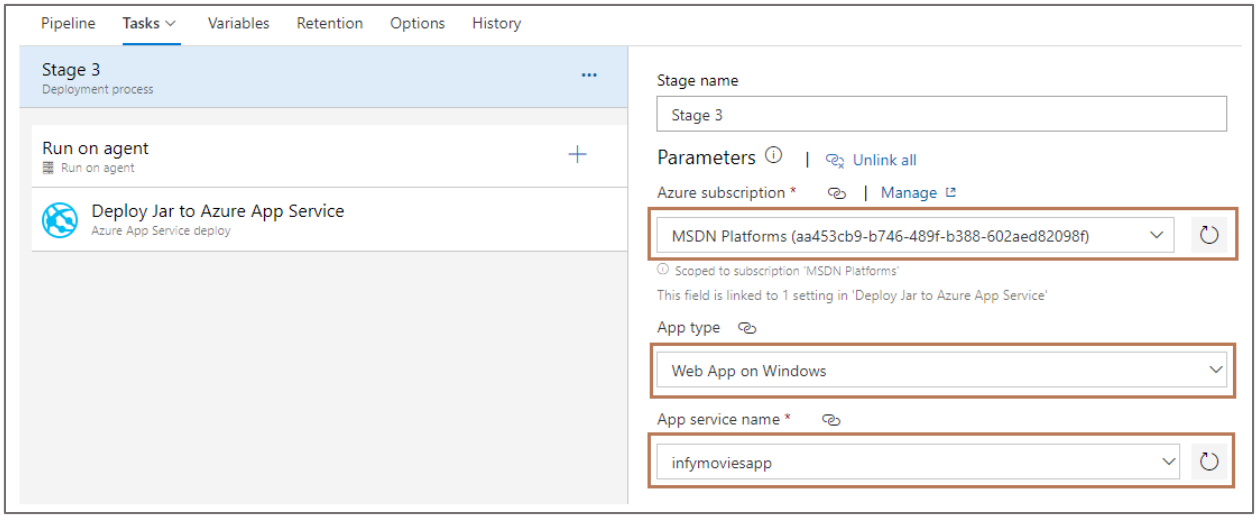
Step 11:

Choose ..

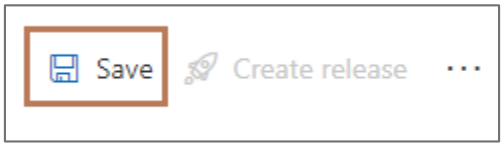

Featured
Azure App Service deployment
Deploy your application to Azure App Service. Choose from Web App on Windows, Linux, containers, Function Apps, or WebJobs.
Deploy a Java app to Azure App Service
Deploy a Java application to an Azure Web App.
Apply

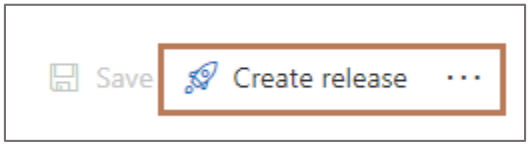
Make sure an App service is already created to deploy the application.

Choose the appropriate subscription, App type and the App service created.

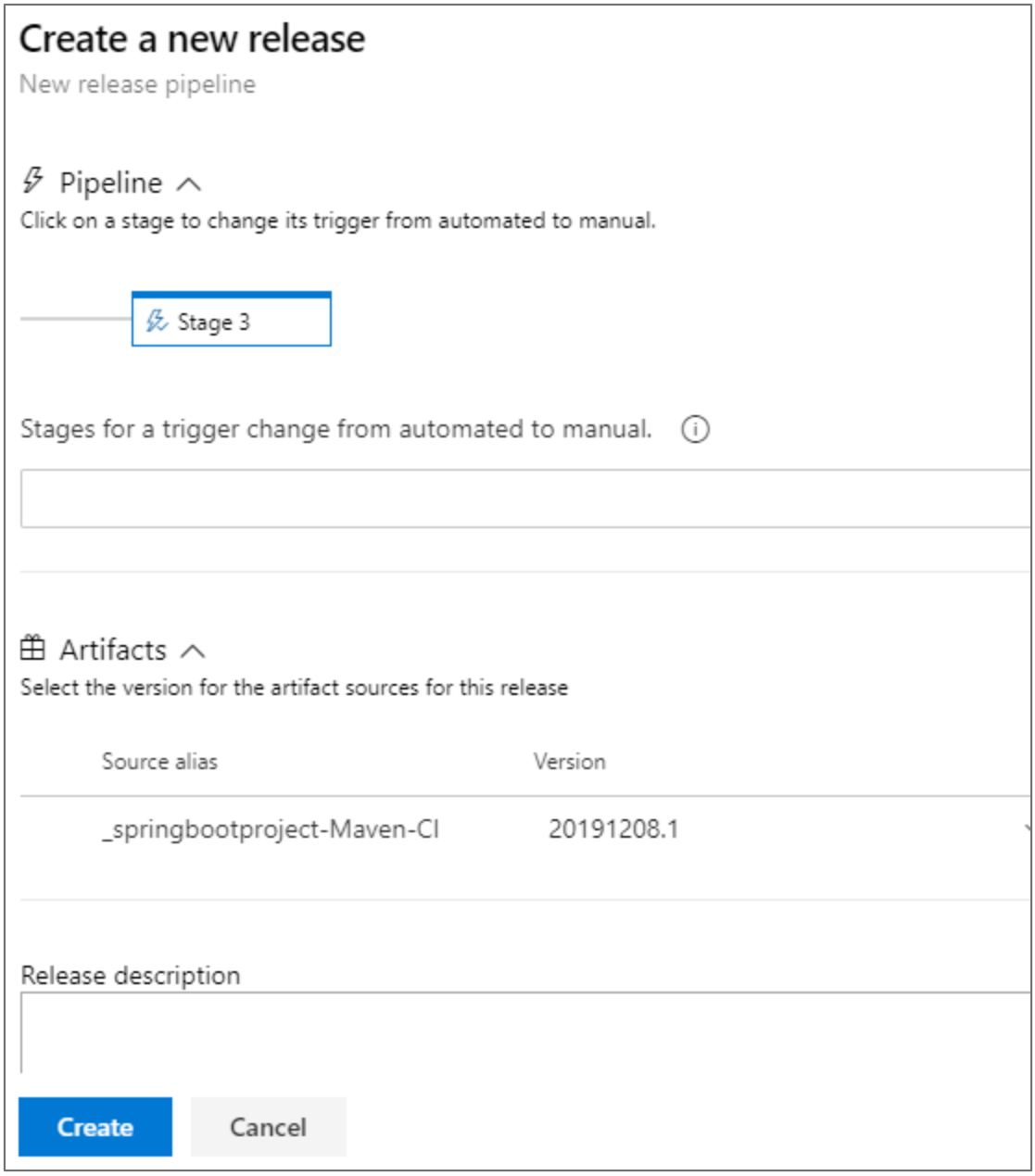


Click on Save and then “Create Release”

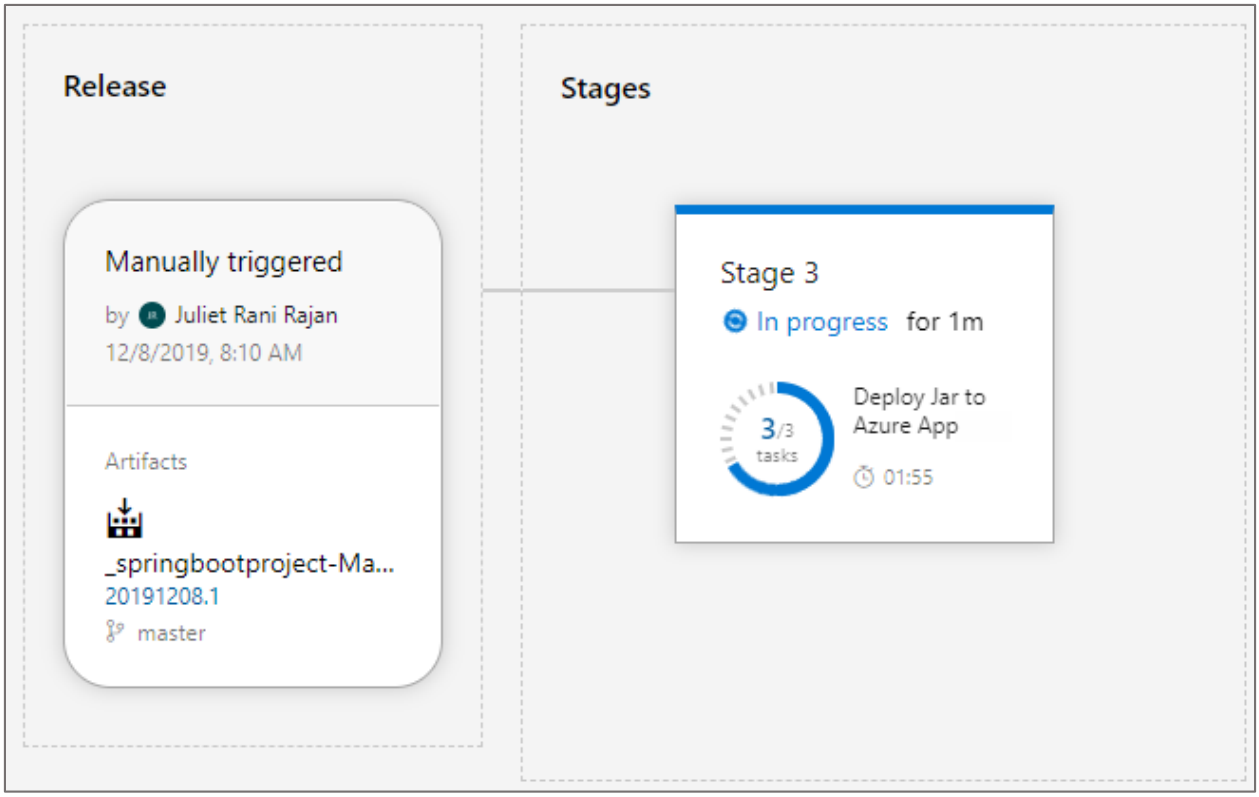




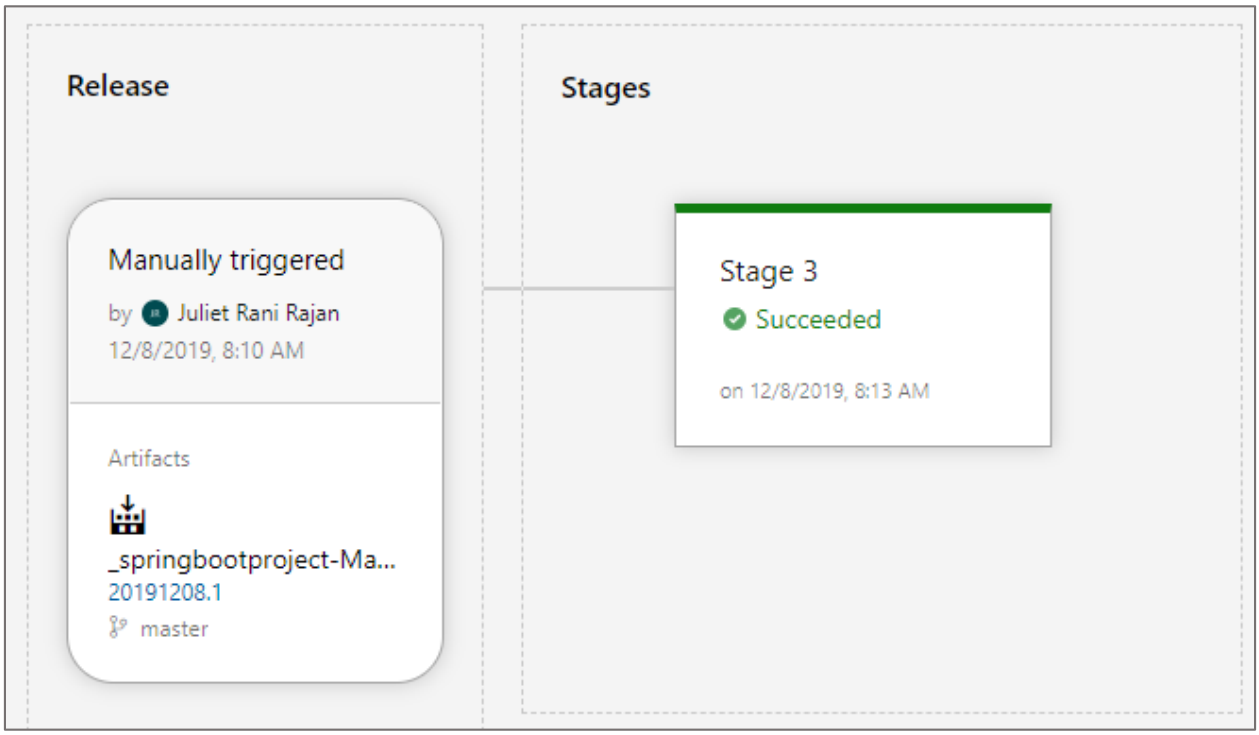
Click on “Create”



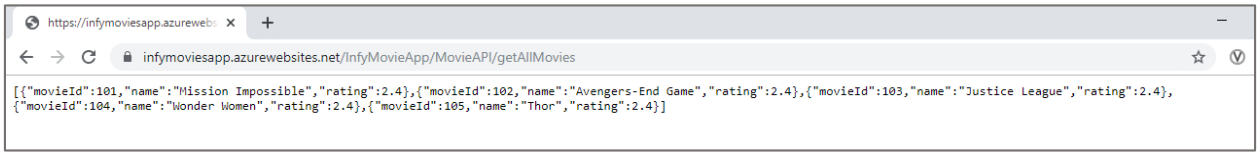
You will see the status of the deployment as below.



On successful, release you will see the below.



You can now access the service.



Thus you have seen how to configure pipeline for java spring boot service.